AUTOMATED INFORMATION DISTRIBUTION IN BANDWIDTH–CONSTRAINED ENVIRONMENTS

Dr. Sam Chamberlain

U.S. Army Research Laboratory
Aberdeen Proving Ground, Maryland, USA

ABSTRACT

In spite of the incredible advances in computing, battlefield information distribution and processing remains archaic because of the common pitfall of simply automating manual techniques. New technologies have been developed to thwart the propagation of this practice and provide automated information distribution between computers using the limited and constantly varying bandwidth of standard combat net radios. The new technologies are based on three major tenets, namely, exchange data (1) in its most general form, (2) only when truly necessary, and (3) in an efficient manner.

INTRODUCTION

The US Army Research Laboratory (ARL) has been exploring concepts and developing technologies to facilitate the exchange of data and information over low bandwidth communications channels. The original goal was to develop a capability to operate successfully over standard tactical radios (VHF–FM) at data rates as low as 1200 bits per second. This requirement remains the standard when evaluating concepts and technologies resulting from this research.

Computer processing power is advancing at an rapid pace with an equally impressive decrease in cost. However, although the future provides an optimistic preview of high bandwidth communication systems, especially in the commercial market, it is expected that the difficult requirements associated with hostile military environments, coupled with budget constraints and legacy communication systems, will typically leave military communication capabilities lagging behind processing power. This is especially true at the lowest echelons between warriors and fighting vehicles. Consequently, *computationally intensive*, rather than *communications intensive* paradigms for command and control (C2) must be developed. This means breaking away from a traditional message–based approach to C2 (e.g., an ‘‘e–mail’’ mentality) to an automated, transaction–based approach that frees the user from the tedious tasks of communications processing.

Under the computationally intensive paradigm, we assume that processing power is infinite in comparison to bandwidth. Every warrior has a data base (i.e., a model) of the battlefield residing in his/her processor, which can be accessed and manipulated by application programs to provide key elements of information required for specific situations. However, the accuracy and synchronization requirements for the data (or information) bases must be flexible to contend with widely varying, often limited, underlying communications capabilities. Whenever possible, a priori information coupled with processing power is used to ‘‘fill in’’ information holes to provide a best guess in the absence of exact information.

To achieve this goal, this program is directed toward three primary objectives:

1. a simple design approach for C2 software;
2. information distribution that is
   a) automated,
   b) adaptive to varying constraints, and
   c) reactive to failures; and
3. a set of general data abstractions of military concepts to serve as the exchange medium.

As seen above, the automatic distribution of information is included as part of this capability, and the technologies to accomplish this are based on three major tenets: exchange data

- in its most general form,
- only when truly necessary, and
- in an efficient manner.

The first tenet, ‘‘in its most general form,’’ includes both C2 schemas and knowledge representation techniques. The first of these addresses the description of the primitive items, activities, or event common to a battlefield. Good examples of this approach are given in [2, 8, 4, 9, 10, 11]. The second concerns the type of storage and retrieval mechanism employed. For example, three approaches to the structuring of data bases are relational, object oriented, and logical (i.e., deductive). Excellent summaries of the last two of these approaches are given in [5] and [1, 6], respectively.

Although the definition of canonical forms of C2 information is a major part of this program, this paper addresses only a few of the storage techniques used to contend with bandwidth–constrained environments. The next section describes a simple but effective software design approach that has been applied to this problem along with an experimental software prototype, called a distributed fact base (or DFB), that has been used to evaluate these concepts. This is followed by a discussion of automated, adaptive, and robust distribution of information in constrained environments. Next, data abstraction techniques are addressed followed by a description of the characteristics of a transaction–based protocol. Finally, a suite of experimental software is described that has been used to evaluate these concepts in several application domains.

C2 SOFTWARE DESIGN APPROACH

A simple software design approach was developed, that to date, has provided a natural decomposition for command and control tasks. All command and control tasks are divided into two very basic categories:

- getting information around, or information distribution, and
- doing something with it once it is there, or battlefield management.

This decomposition is illustrated in Figure 1.

The *battlefield management* task consists of the myriad complex and sophisticated operations that must be executed to win a battle. The *information distribution* task ensures that information is available at the many locations at which the battlefield management

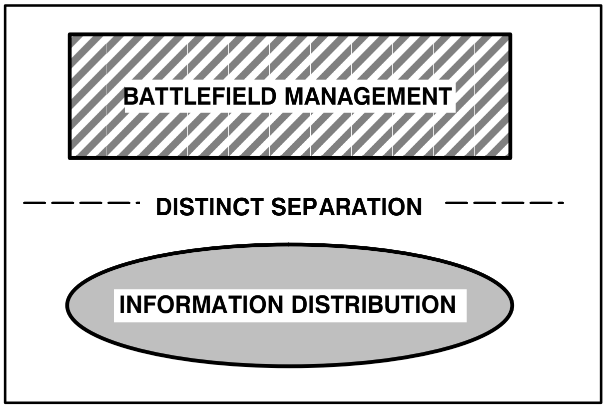

Figure 1: A Simple C2 Task Decomposition

tasks reside. Because this decomposition is applicable to systems other than battlefield management, we often give the battlefield management tasks the more generic title of *application programs*. A distinct separation exists between the information distribution and battlefield management tasks, and there is a many–to–one relationship between application programs (battlefield management tasks) and the information distribution task [7]. In other words, there is one information distribution task per node with many application programs attached to it.

A key design feature is that the information distribution task combines the functions of *data storage* (e.g., data bases) and *communications*. Application programs connect directly to the data storage function via a clearly defined interface specification. The data storage functions at each node are then connected to each other via the communication function, totally isolated from the application programs. This means that the application programs need know nothing about communications; from their perspective, they are connected to a data base from which they collect and add information – the communications between data bases remains hidden.

A major thesis derived from this software design approach is that a single information distribution system can serve *all* battlefield management functions, regardless of the military service, branch, or nation provided that

A. common data are used (e.g., data abstractions of military concepts),
B. necessary services are provided to the application programs to allow them to truly divorce themselves from the information distribution task, and
C. worst case communications are handled.

All three of these factors are addressed in this research and this thesis drives the long–term goals and objectives for this program.

DISTRIBUTED FACT BASES (DFB)

To evaluate the information distribution concepts just described, an experimental software prototype (i.e., ‘‘our workbench’’) has been implemented. Called a *distributed fact base*, or *DFB*, this software implements the information distribution task. Conceptually, a DFB exists at each (mobile) node along with mission–specific application programs. The link between application programs and their DFB is via standard DoD TCP/IP (Transmission Control Protocol/Internet Protocol) sockets [12], [13], although any connection–oriented protocol is equally applicable. But while ‘‘local’’ application programs communicate with the DFB via relatively reliable, high speed communications link, the inter–DFB communications are considered to be potentially unreliable, low bandwidth communication links, such as tactical radios. Therefore, suitable communication protocols must be developed with features to contend with this environment (e.g., the Fact Exchange Protocol, or FEP, described later, is one such example). This configuration is illustrated in Figure 2.

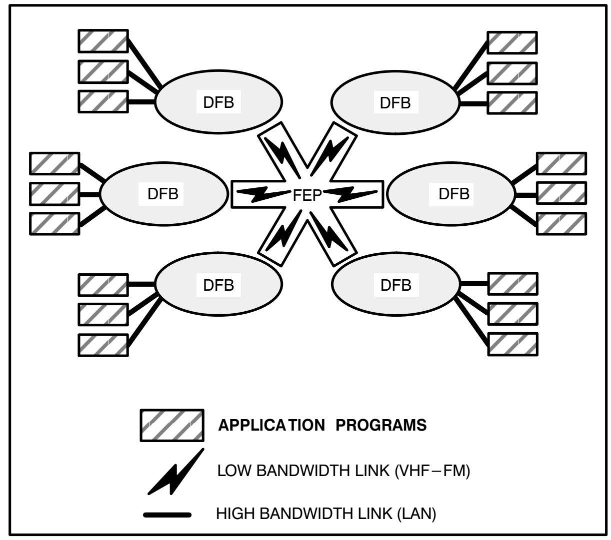

Figure 2: Several DFBs with Applications

AUTOMATED INFORMATION DISTRIBUTION

In any C2 system, the ultimate goal is to transfer concepts and ideas between human beings. More and more, this transfer is being extended to include computers. The problem with current systems is that a human is the arbiter for exchanges between computers, normally as a reviewer of a message queue. This tedious task places undue pressure on the operator as well as producing a bottleneck in the system. The goal for the future battlefield is to allow computers to exchange information directly (between data bases, using data abstractions rather than messages) without the requirement of human intervention; this concept is illustrated in Figure 3. Note that this in no way implies the demise of voice, image, text, video and other human–oriented forms of communications.

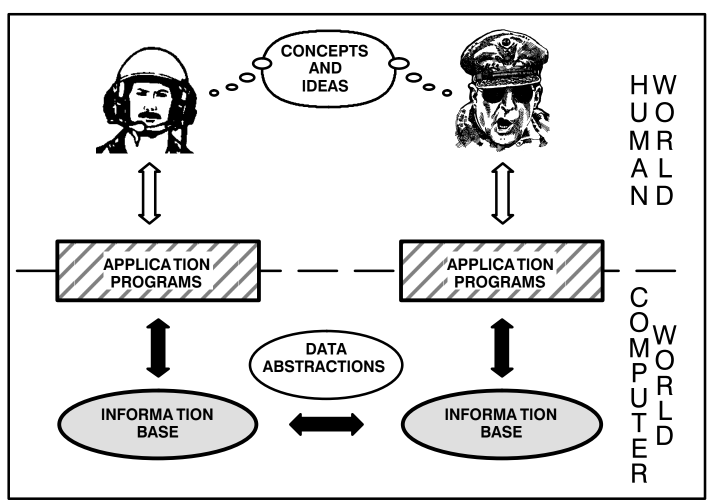

Figure 3: Data Abstractions as the Medium for Data Exchange

Automated information distribution, like all C2 functions, requires an equal emphasis on both *computer science* and *military science*. In this case, the military science aspects are the description of

- the military concepts that are to be stored and exchanged (i.e., the data abstractions of military concepts), and
- the criteria that determine what, when, and to whom information is worthy of transmission, and how the exchange is to be accomplished.

Using this paradigm (see Figure 3), computers are used at the end points to convert human concepts into abstractions. The abstractions are then exchanged between computer information bases. Upon reception at the end points, computers are again used to convert the abstractions back into human concepts, which are presented in a manner appropriate for the operator and situation (e.g., a symbol on a heads–up display). This removes the human from the tedious task of ‘‘reading’’ messages and also removes one of the major bottlenecks from the system.

The second tenet (provided earlier) for exchanging information in constrained environments was ‘‘to exchange information only when necessary.’’ In current C2 systems, this is determined manually, on a case–by–case basis, by a human operator. This is often not acceptable, especially as one moves down the echelons toward the fighting individual or crew. In fact, contrary to the popular belief of many, the automation of information distribution becomes more important at the lower echelons. This is because it is not the volume of data that causes trepidation, but the high priority of other tasks that the individual must accomplish. Simply put, a soldier trying to fight and survive a battle has little time to type information into a keyboard. Consequently, it is imperative that the information exchange facilities at these echelons be automated.

To implement such a scheme, techniques must be developed to describe the *criteria* to direct information exchange. In essence, this is nothing but standard operating procedures (SOP) for communications. As part of this research, *information distribution commands (IDC)* have been developed that possess the capability to describe when, what, and where, and how information should be exchanged. Each command has a criteria part and an action part. A typical example could be ”when information about four or more armored vehicles arrives (criteria), then send that information to my parent and adjacent nodes (action).” Further, to expand the descriptive power of the criteria portion of the IDC, supplementary information is maintained about the current and last transmitted value for each data item, the source and destination of the information, and whether a data item even exists.

Any time information enters a DFB, whether from another DFB or an application program, the IDC criteria are searched for matches. If a match occurs, three actions may occur. First, information may be sent to another DFB. An IDC with this action is called a *distribution rule*. Second, an application program may be notified of the arrival. IDCs with this action are called *triggers*. Application programs may insert triggers into the DFB; hence, they have two ways of obtaining information from a DFB: manually via queries or automatically via triggers. Finally, overheard information (i.e., information not addressed to this particular DFB and not broadcast to all) may be entered into the DFB. Information overhearing is a major portion of the strategy for effective operation in bandwidth–constrained environments. Thus, the IDCs provide a general capability to describe the handling of information by the DFB.

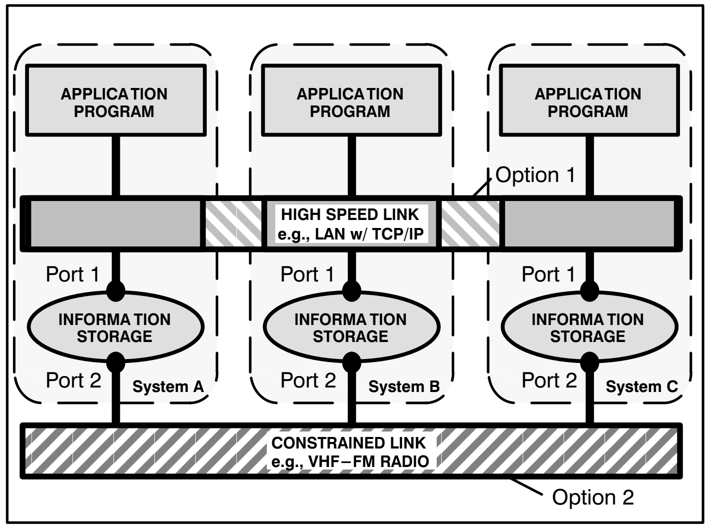

Figure 4: The Two Access Ports of a DFB

Because the same IDC may reside in several nodes (normally in sibling units), infinite loops or redundant copies of information updates can easily propagate. For example, adjacent units with identical IDCs can bounce the same information back and forth forever. To prevent this situation, four overriding criteria are always in effect:

I. Never send a message back to its source.
II. If a message is from a parent, do not send it to an adjacent unit.
III. If a message is from an adjacent unit, do not send it to anyone.
IV. If a message is from a sub–unit, do not send it to another sub–unit.

These four rules prevent information looping caused by the same IDC at several nodes and allow an innovative new information distribution policy. Briefly stated, *when passing information up the hierarchical organization tree, also include any sibling and adjacent units that require the information*. This scheme propagates information as a breadth–first “wave” up the echelons, allowing the units that are most affected by the information to obtain it first. This approach is contrary to typical doctrine that requires information to flow up and down a rigid tree structure when adjacent nodes are not in the same command (i.e., have the same parent). However, time–critical, spatial information, such as sightings of enemy or unknown vehicles for fratricide prevention, adversely affects adjacent units and should be passed directly to them. Thus, a tank section may very well send data directly to adjacent tank sections even though they belong to different tank companies, corps, or nations.

A DFB has two portals of entry: one via a high–speed, connection–oriented port for application program interface with access by queries and triggers and a second whose access is strictly controlled by the distribution rules because it may be connected to a constrained link. If a high–speed link is available between two nodes, then there are two connection options: one via the connection–oriented port, or another via the constrained link port; this is illustrated in Figure 4. The option selected depends upon the control required for the particular situation. The point is that high–bandwidth links do not present a problem other than challenge the assumption that processing power is infinite relative to bandwidth.

The previous paragraph introduces the concept that distribution rules must be adaptive to varying bandwidth capabilities.

Adaptive information distribution is being accomplished by maintaining networks statistics via passive overhearing of the single hop network to which the DFB is connected. Statistics are also maintained about the length of time an outgoing data base update remains in the queue and on the round trip time for acknowledgments. From this information, network throughput, delay, connectivity, and other information can be calculated, thus providing a model of the network. This information is then added to the data base, updated periodically and therefore can be used by the IDCs just as any other information. Thus, the condition of the network, such as average delay or throughput, becomes part of the distribution rules. This allows information to be prioritized based on the condition of the network. For example, if the network delay is low, then 100–meter location data may be exchange. As the delay increases, the accuracy can be decreased to 500 meters and so on. During periods of high congestion, the exchange of location information may be halted. On the other extreme, when a high bandwidth link is available, concurrent data bases can be established.

One other topic being addressed is *robust*, *automated information distribution*. Bear in mind that there is no ''send button'' under this paradigm. The user is not directly involved in the distribution process. So the question then arises, what does one do in a totally automated system when the system fails? The approach to address this issue is twofold. First, the IDCs will be modified by adding an ''else'' part that describes what to do for a failure (e.g., **if** this occurs **then** send this information **else** [if it fails] do this). This is an interesting problem because the reaction to a failed rule depends on the situation described by the rule—there is no standard answer (to notify the operator and let him/her worry about it is not an acceptable solution). Further, a rule may fire as a result of data inputs by several application programs; thus, notifying a single application program is not sufficient. Second, the network statistics gathered will be used to make recommendations about the problem (e.g., connectivity information can provide insight as to whether a node has failed) and eventually, a recommended solution (e.g., check your own radio antenna).

There are several subtle ramifications of using IDCs to control information distribution:

- An operator never has to explicitly send data; this is determined by the IDCs each time the data base is updated. (The data base may be updated by the operator via an application program or directly by another system, e.g., a position location device.)
- If the IDCs are thoughtfully developed (ahead of time), then one should *never* have to query another node! (Queries are very expensive in communications bandwidth.)
- Information cannot leave a node unless an IDC fires. This provides a *security policy* for the node and imparts *discipline* on digital communications.
- IDCs provide data resolution adjustments up/down the command chain—higher echelons get lower resolution information.

Finally, the successful automation of information distribution requires that two other obstacles be conquered. The first is trust by the user. The application of computers to real–time battlefield command and control is new. It will take time to gain the trust and confidence of the soldier who must depend on these systems for survival. This problem can be attenuated by providing the operator with simple and timely feedback concerning the status of information exchange process. (This research is currently addressing this area.) The second is that IDCs (i.e., formal communications SOPs) must be defined. This is a military science problem (some might even call it doctrine) that has rarely been explicitly considered. Command and control requires equal parts of military and computer science, and this problem may provide the impetus for a re–evaluation of the partition of roles between the combat and materiel developer.

EFFICIENT INFORMATION TRANSFER

To transfer digital information efficiently in a battlefield environment, a protocol was developed that exploits the characteristics of standard combat net radio (CNR) channels, i.e., unreliable, low–bandwidth, broadcast communications. The major problem of these networks is congestion, and consequently, large, outgoing message queues and delays.

The Fact Exchange Protocol (FEP) is a reliable datagram protocol that supports multicast and broadcast addressing, message concatenation, and ''overhearing,'' i.e., all information exchange on the network is collected whether it is addressed to that node or not [3]. However, the most unique feature is the ability to postpone the building of an outgoing packet until the channel is clear. This ensures that the most important information always gets priority because it does not have to wait for previously queued information. This feature is possible because of the relatively slow speed of the channel in comparison to the speed of the computer and because the FEP is design to respond to signals from the datalink layer services (this layer determines if the channel is clear in a carrier sense, multi–access protocol).

ABSTRACTION OF MILITARY CONCEPTS

The 1990's mark the progression into the *information age*. For the first time, information is becoming readily available to the masses and at an accelerating rate. However, myriad storage techniques and structures have also proliferated during this evolution, and military command and control systems are no exception. This uncontrolled proliferation combined with the practice of automating manual techniques has resulted in the absence of interoperability between battlefield C2 systems. As previously stated, this paper does not address C2 schemas but presents several techniques incorporated to support the distribution of information over bandwidth–constrained environments.

Each DFB includes a RAM resident storage facility for information named facts. Facts correspond to objects in object–oriented data bases, tuples in relational data bases, and unconditional ground clauses in deductive data bases. Every time a new fact is created (i.e., stated to a DFB), a universally unique *fact identification number* (or fact–id) is assigned and is returned to the originator of the fact; this is illustrated in Figure 5. The fact–id becomes an explicit part of the data and because of its universal uniqueness, a fact–id can be used across DFBs just as pointers are used within computer programs; this is what makes a DFB ''distributed.''

The fact structures are named fact–types and define what is commonly called the data dictionary. Fact–types correspond to relation schemes in relational data bases and classes in many object–oriented data bases. Finally, the attributes of a fact–type are named fact–items. The domain of fact–items currently includes integers, floating point numbers, character strings, references (fact–ids), and lists of fact–ids (lists). Thus, *fact–types* describe the structure of the facts while *facts* are instances of the fact–types and each has a universally unique fact–id; each fact is composed of *fact–items* that may include one or more fact–ids as values.

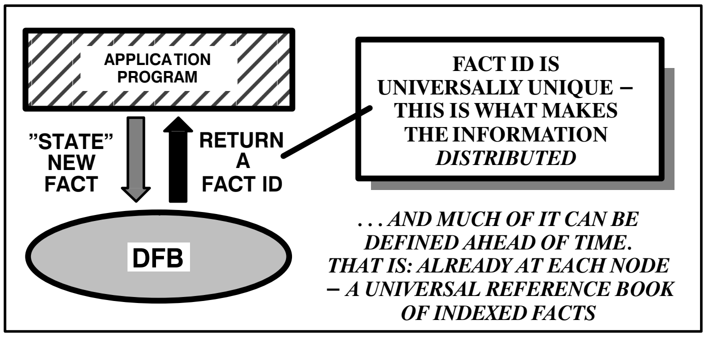

Figure 5: Creation of Facts and Fact IDs

Because fact ids may be used as pointers to other facts, this structure contains many of the features associated with object–oriented programming (e.g., inheritance). The major difference is that the linking information (i.e., pointers) is defined explicitly, as part of the data, rather than being independently created by each machine. This allows the information base to be truly distributed, and because fact–ids can be freely exchanged in lieu of more voluminous forms of data, unnecessary information exchange between data repositories can be minimized. Further, levels of indirection (e.g., following the pointers) can be used to create more efficient data structures analogous to using pointers in a programming languages. Fact–ids also make the data independent of any particular natural language. For example, if a fact–id references a vehicle fact describing a U.S. M1A1 Abrams tank, it does not matter if the description is in French, English, or Japanese. Identical fact–ids reference identical semantic entities.

In this approach, information is divided into three categories: *reference material* that everyone should have ahead of time; *dynamic information* that is created, destroyed, and exchanged throughout a battle; and *meta–information* about the information itself. Reference material is a source of a priori information (mentioned earlier) that can be used to fill information holes caused by limited communications bandwidth. It includes information about the characteristics of organizations, its equipment and personnel, and other data from reference manuals. This is information that can be created ahead of time, and consequently, can have universally defined, static fact–ids. The word static does not imply that the data inside the fact are static (although they may be), but only that the existence of the information is static. Examples are vehicle types and characteristics and standard weapons suites and capabilities.

Finally, facts serve as the basis of information transfer between DFBs. In other words, the traditional ‘‘message’’ is replaced by *fact exchanges*, which are direct data base updates. A key advantage of this is flexibility. Any single fact–item of a fact can be updated, thus predefined message formats composed of fixed fields are replaced by a mechanism that allows only required data to be exchanged. When combined with reference material (i.e., predefined facts with universally known fact–ids) this both minimizes the amount of data traffic required to exchange concepts while also providing the foundation to build the IDCs (e.g., distribution rules).

SUMMARY and CONCLUSIONS

An approach for designing command and control software has been presented that blends the data storage and communications tasks to provide an automated information distribution function. Application programs connect directly to the data storage portion of this function via a clearly defined interface specification. The data storage tasks at each node are connected to each other via the communication task, thus totally isolating application programs from tedious communication tasks.

Three tenets were introduced that address the goal of implementing automated information distributed in bandwidth-constrained environments. The first tenet is to send information in its most general form and requires the re–evaluation of basic military concepts with the ultimate goal of abstracting them into a canonical form ideal for processing by machines. The second tenet addresses automated information exchange. Information distribution commands are developed that describe thresholds that indicate when information warrants exchange; in other words, realistic synchronization requirements for the data bases that include information concerning the condition of the communication links between the data bases. This allows synchronization requirements to be defined that adapt to the available bandwidth. The third tenet focuses on network protocol issues. Features that have long been used in voice communication can be applied to digital communication (e.g., overhearing the data base update to others to obtain free information). By postponing the building of outgoing packets (i.e., the selection of outgoing data base updates) until the last possible moment, the most important updates always leave first (as opposed to waiting in a queue on a first–come first–served basis).

The concepts that surround the three basic tenets have been evaluated via experimental software prototypes called distributed fact bases. However, to accomplish automated command and control, doctrine and materiel developers must work together, perhaps redefining the boundaries between these responsibilities, to push beyond the simple automation of manual techniques into the realm of true battlefield automation that is accepted as an asset by the warrior.

REFERENCES

[1] J. Grant, J. Minker, ‘‘The Impact of Logic programming on data bases,’’ *Communications of the ACM, Vol 35(3)*, March 1992.
[2] G. Hartwig, Jr., ‘‘The Information Distribution System: The Factbase,’’ BRL Technical Report BRL–TR–3247, July 1991.
[3] V. Kaste, ‘‘The Fact Exchange Protocol. A Tactical Communications Protocol,’’ BRL Memorandum Report BRL–MR–3856, August 1990.
[4] I. Mayk, ‘‘The Command and Control Reference Model,’’ Basic Research Group of the JDL Tech. Panel for C3 (JDL TPC3), TR–94–1, 20 Jan 1994.
[5] I. Mayk, J. Kornell, A. Fastlich, ‘‘Toward C2ED_Object Libraries,’’ Proc. 1993 JDL Symp. on C2 Research, National Defense Univ., Wash, DC., Jun 1993.
[6] J. Robinson, ‘‘Logic and Logic Programming,’’ *Communications of the ACM, Vol 35(3)*, March 1992.
[7] K. Smith, ‘‘The Information Distribution System: Aplication programmer's Interface Guide,’’ BRL Memorandum Report BRL–MR–3902, April 1992.
[8] ‘‘The ASDC3I/JIEO Fire Support Data Model,’’ August 1993.
[9] ‘‘The Joint Warfare Simulation Object Library – Requirements Definition,’’ NCCOSC RDT&E Division, Code 42, 7 February 1994.
[10] ‘‘The NATO Army Tactical Command & Control Information System (ATCCIS) Battlefield Generic Hub Data Model,’’ 23 April 1993.
[11] ‘‘The US DISA/JIEO Command and Control (C2) Core Data Model,’’ 1 September 1993.
[12] US Military Standard 1778, ‘‘Transmission Control Protocol,’’ DDN Protocol Handbook, *Vol 1*, December 1985.
[13] US Military Standard 1777, ‘‘Internet Protocol,’’ DDN Protocol Handbook, *Vol 1*, December 1985.